Increasing the expression power of persons' profiles in semantic social networks

Puustjärvi, Juha

2017

Puustjärvi , J & Puustjärvi , L 2017 , ' Increasing the expression power of persons' profiles in semantic social networks ' , Procedia Computer Science , vol. 111 , pp. 8-16 . https://doi.org/10.1016/j.procs.2017.06

http://hdl.handle.net/10138/231215
https://doi.org/10.1016/j.procs.2017.06.003

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 111 (2017) 8–16

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Advances in Information Technology, IAIT2016, 19-22 December 2016, Macau, China

Increasing the expression power of persons' profiles in semantic social networks

Juha Puustjärvi[a,]*, Leena Puustjärvi[b]

[a]*Department of Computer Science, University of Helsinki, P.O. Box 68, Helsinki, Finland*
[b]*The Pharmacy of Kaivopuisto, Neitsytpolku 10, Helsinki, Finland*

Abstract

The amount of social information published in the Web has dramatically increased in the past years. FOAF, Dublin Core, and vCard are examples of popular vocabularies that are used to present computer understandable profiles of persons in social media. These profiles describe facts about a person such as his or her interests, and thereby we can easily find persons having the same interests. However, in reality we are rather interested in about the closeness of persons' interests than whether they have precisely the same interests. Such aspects are easily understandable for humans but they are not machine understandable without additional semantics. Therefore we have extended the vocabularies of social media by domain specific taxonomies. In addition, to increase the expression power of persons' profiles we have also attached weights to persons' interests. We have also consider a variety of ways for expressing and computing the closeness of weighted profiles.

Peer-review under responsibility of the organizing committee of the 8th International Conference on Advances in Information Technology

Keywords: Semantic social network; Semantic web; FOAF; Ontologies; OWL; RDF ;

1. Introduction

A social network is a collection of individuals linked together by a set of relations[1]. A semantic social network is the result of the application of Semantic Web technologies to social networks[2]. For example, several vocabularies

* Corresponding author. Tel.: 358452756720.
E-mail address: juha.puustjarvi@cs.helsinki.fi

1877-0509
Peer-review under responsibility of the organizing committee of the 8th International Conference on Advances in Information Technology
10.1016/j.procs.2017.06.003

such as FOAF (friend of a friend)[3], Dublin Core [4], and vCard[5], are used in social networks. These vocabularies can be used in describing facts about a person such as his or her work place, job title and interests.

FOAF is a machine-readable vocabulary describing persons, their activities and their relations to other people and objects. Anybody can use it to describe him- or herself. It is expressed using the Resource Description Framework (RDF)[6] and the Web Ontology Language (OWL) [7]. Computers may use these FOAF profiles to find, for example, all people interested in tennis, or to find out all people both you and a friend of yours know.

Technically, FOAF vocabulary (ontology)[8] is comprised of things and links. Links are called properties and the types of things are called classes. Hence, FOAF vocabulary is comprised of terms, each of which is either a class or a property.

Class foaf:Person is one of the most used class of the FOAF vocabulary. A set of properties, such as gender, birthday, interest and topic_interest, can be attached to the instances of this class. Hence, we might claim that a person, say John, has an interest in RDF by saying that "John stand in an interest relationship to RDF". Likewise we might claim that another person, say Mary, is interested in Semantic Web.

John might be interested in whether he and other persons, such as Mary, do have interests in common? Individuals familiarized on these topics, such as John and Mary, understand that they do have some interests in common as John is interested in RDF and Mary is interested in Semantic Web. However, computer cannot make such a conclusion without machine interpretable knowledge about the dependency of the terms RDF and Semantic Web. Hence, machine understandable data is of prime importance as the amount of data in social media is increasing all the time.

Using the FOAF property topic_interest John might also state that he has an interest in classic blues and punk. However, he cannot state that he is extremely interested in classic blues and has minor interest in punk, as such expressions would require some kind of weighting of interests. Yet, ignoring weighting considerably restricts the expression power of persons' profiles.

In this paper, we will restrict ourselves on the above mentioned inconveniences:

- First, we deal with taxonomy-based semantic similarity, i.e., how the semantics of taxonomies can be exploited in computing the closeness of persons' interests. In particular, we adapt the methods of genealogy for computing the closeness of persons' profiles.
- Second, we consider how weighting can be used in making profiles more specific, and how the closeness of such profiles can be computed. First we extend the taxonomy-based semantic similarity by weighting, meaning that a person may attach weights low, medium or high to his or her interests. Then, we consider the usability of the Vector Model[9] for computing the closeness of persons' interests. The key idea behind this model is that persons' interests as well as queries can be presented as vectors. Hence, we can process a query by computing the distance of the query vector and the profile vectors.

The rest of the paper is organized as follows. In Section 2, we first consider the structure and semantics of taxonomies, and then we present how taxonomies can be used in extending the expression power of the FOAF vocabulary. In Section 3, we give an example of a FOAF file, and illustrate how FOAF files are queried by SPARQL[10]. Taxonomy-based semantic similarity is the topic of Section 4. Then, in Section 5, we deal with profile weighting in the context of taxonomy-based similarity, and in the context of the Vector Model. We also shortly consider the matching algorithms that are suitable for the Vector Model. Finally, Section 6 concludes the paper by discussing our future research.

2. Taxonomies

2.1. The structure and semantics of taxonomies

One of the main challenges faced by information retrieval systems is to efficiently manage the large number of documents they hold. Such systems make it easier to give users access to relevant resources that satisfy their information needs. Information retrieval systems are based on an information retrieval model. They determine the ways the data of the resources are presented, and the way the information needs (queries) are presented.

FOAF deals with the representation of FOAF files, which are collections of facts about a person such as where they work, where they went to school, and their friends are. Each fact represents a property and its value attached to a thing such as to a person.

Each fact is presented by an RDF-statement, i.e., by subject–predicate–object expressions[8]. These expressions are known as triples in RDF terminology. The subject denotes the resource, and the predicate denotes an aspect of the resource and expresses a relationship between the subject and the object. Such triples are also often understood as the thing being described, its property name, and property's value[11].

For example, the notion "John has interested in tennis" can be presented by a triple where the subject is "John", the predicate is "has interest", and the object is "tennis". That is, the predicate represents the property attached to a thing.

In order to standardize the values of properties (predicates), specific ontologies are introduced in many disciplines. An ontology provides a general vocabulary of a certain domain, and it can be defined as "an explicit specification of a conceptualization"[12]. In essence, an ontology gives the semantics to the metadata[13]. Typically such ontologies are hierarchical taxonomies of terms describing certain topics[14]. For example, the topics of the taxonomies in Fig. 1 are ICT, Healthcare, and Music.

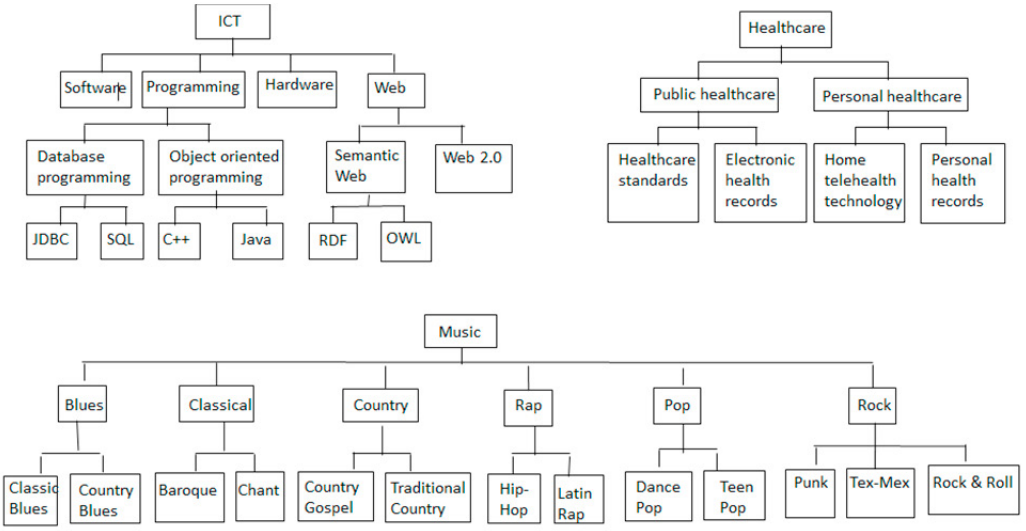

Fig. 1. Examples of taxonomies.

The logic behind taxonomy is that when we go up the taxonomy toward the root, the items become more general, and respectively when we go down towards the leaves the items become more specialized. We can also state this in a more formal way: depending on the direction of the link each link between a parent and a child node represents a *subclassification* relation or *superclassification* relation[13]. For example, in the ICT-taxonomy Semantic Web is a *subclassification* of Web, and Web is a *superclassification* of Semantic Web.

2.2. Extending FOAF by taxonomies

The description of the terms in the FOAF 'dictionary' often makes reference to classes and properties elsewhere [8]. For example, in the context of FOAF we might say in OWL that a FOAF property has a domain or range of an externally defined class. These kind of claims help fix the intended meanings of FOAF terms in relationship to other vocabularies, e.g., to taxonomy.

To illustrate this, an ontology, called Taxonomy ontology, is presented in Fig. 2.

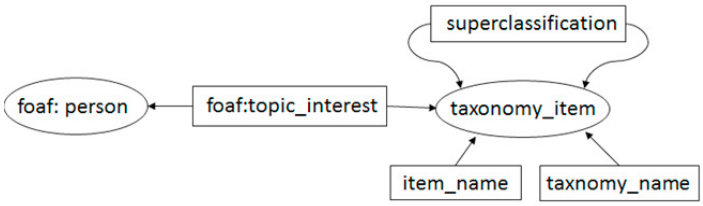

Fig. 2. Taxonomy ontology standardizes the values of the foaf:topic_interest property.

In this figure ellipses represent classes and rectangles represent *data type* and *object properties*. *Classes*, *data properties* and *object properties* are modeling primitives in OWL. Object properties relate objects to other objects while datatype properties relate objects to datatype values. This graphical OWL ontology states that the values of the property *foaf:topic_interest* are the items (nodes) of one or more taxonomies. Hence, we might specify that the allowable values of the *foaf:topic_interest* are the nodes of the ICT-taxonomy.

For example, the following file states, in Turtle format [15], that Web and Web 2.0 are items of the ICT-taxonomy and the former is the *superclassification* of the latter.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix ict: < http://www.cs.helsinki.fi/u/puustjar/ict_taxonomy#>.
<#Web>
  a ict:taxonomy_item;
<#Web 2.0>
   a ict:taxonomy_item;
   ict:superclassification <#Web>
```

Fig. 3. Inserting two nodes into ICT-taxonomy.

3. FOAF Files

3.1. Representing interests in FOAF files

Anyone can use FOAF to describe his or her profile by a FOAF file[16]. To illustrate this, let us consider the following FOAF file (written in Turtle format). It states that John Taylor is the name of the person described here. His e-mail address, homepage and depiction are web resources, meaning that each can be described using RDF as well. He has Wikipedia as an interest, where Wikipedia is identified by its homepage. He has also three topics of which he has interest in. The predicate of these RDF-statements is *foaf:topic_interest*, and the values of these predicates are “Java”, “RDF” and “Personal health records”.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix ict: < http://www.cs.helsinki.fi/u/puustjar/ict_taxonomy#>.
@prefix health: < http://www.cs.helsinki.fi/u/puustjar/health_taxonomy#>.
<#John>
a foaf:Person;
   foaf:name "John Taylor";
   foaf:mbox <mailto:jt@mail.com>;
   foaf:homepage <http://www.johntaylor.com>;
   foaf:nick "JT";
   foaf:depiction <http://www.johntaylor.com/img.jpg>;
   foaf:interest <http://www.wikipedia.org>;
   foaf:topic_interestict ict:Java ;
   foaf:topic_interest ict:RDF;
   foaf:topic_interest health:Personal health records.
```

Fig. 4. John Taylor’s FOAF profile in Turtle format.

In this FOAF profile the properties *foaf:interest* and *foaf:topic:interest* is used for stating Joh's interests. These properties are similar in the sense that (according to FOAF vocabulary) their Domains are Agent. The Agent class is the class of agents; things that do stuff. Its commonly used sub-class is *Person*, representing people. Other kinds of agents include *Organization* and *Group*.

The properties *foaf:interest* and *foaf:topic:interest* differ in that the Range of the former is Document, whereas the Range of the latter is Thing. The Document class represents those things which are broadly conceived as documents (e.g., the document at http://www.wikipedia.org> of John's FOAF profile in Fig. 4).

OWL has introduced the class 'Thing' as a name for the universal class of all things. This is useful as we want to express universality of property use, e.g., that anything can be the value of *foaf:topic_interest*. In our case its values are the items of taxonomies, e.g. the items of ICT-taxonomy, Healthcare taxonomy or Music taxonomy.

In our used terminology, by a *foaf:topic_interest profile* of a person we refer to its values. So, for example, John's *foaf:topic_interest profile* is the set comprising of Java, RDF, and Personal health records.

3.2. Querying FOAF Files by SPARQL

One can query FOAF files by SPARQL. The name SPARQL is a recursive acronym for SPARQL Protocol and RDF Query Language[16]. A typical SPARQL query specifies the pieces of information that meets the given conditions. The conditions are described with triple patterns, which are similar to RDF triples but may include variables to add flexibility in how they match against the data. There is also a variety of SPARQL processors, which are also called SPARQL engines, available for running queries against data both locally and remotely.

For example, assume that John is willing to know persons having similar profiles. John is interested in Java, RDF and Personal health records. He can then query profiles that have all these interests or some of these interests. The latter query is presented in SPARQL in Fig. 5.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ict: < http://www.cs.helsinki.fi/u/puustjar/ict_taxonomy#> .
@prefix health: < http://www.cs.helsinki.fi/u/puustjar/health_taxonomy#> .

SELECT ?person
WHERE
{
      { ?person foaf:topic_interest ict:Java . }
        UNION
      { ?person foaf:topic_interest ict:RDF . }
        UNION
      { ?person foaf:topic_interest health:personal health records . }
 }
```

Fig. 5. Querying persons having interests in Java, RDF, or Personal health records.

In order to illustrate the result of this query assume that Mary has a FOAF file, and her *foaf:topic_interest* profile includes taxonomy items "OWL" and "Personal health care". As John's and Mary's profiles do not include any common taxonomy items, Mary's profile will not be included in the result of John's query presented in Fig. 5. This is regrettable as they still have common interests:

- John is interested in RDF and Mary is interested in OWL, and in the ICT-taxonomy (Fig. 1) they both are *subclassifications* of the taxonomy item "Semantic web".
- John is interested in "Personal health records" while Mary s interested in Personal health, which is the *superclassification* of "Personal health records" in the Healthcare taxonomy presented Fig. 1.

4. Taxonomy-Based Semantic Similarity

In *genealogy*, also called as family history, the closeness between two persons is comparable with the proximity of two family members in family tree: the more two members of the family have ancestors in common, the closer they are[17]. The distance of a family member starting from a common ancestor determines his or her distance compared to the other family members. The closeness of two persons is thus a question of relationship between the numbers of the common ancestors of these persons[18].

We have adopted this similarity measure for computing the closeness of persons' interests. For example, let us consider the concepts C++ and Java in ICT-taxonomy (Fig. 1). The number of their common ancestors is 3. Correspondingly concepts C++ and SQL have 2 common ancestors.

We denote the number of common ancestors of two concepts (taxonomy items), say Ci and Cj by the notation CommonAncestors(Ci, Cj). So, for example, CommonAncestors(C++, SQL) = 2.

We determine the similarity measure of the concepts Ci and Cj as follows:

If Ci = Cj, then Similarity (Ci, Cj) = 1; otherwise

$$Similarity(Ci, Cj) = \frac{CommonAncestors(Ci,Cj)}{max\{RootDistance(Ci),RootDistance(Cj)+1\}}$$

where *RootDistance(C)* denotes the distance (in the number of links) between the concept *C* and the root in the taxonomy. The function of the denominator is to normalize the value of *Similarity(Ci, Cj)* to be less or equal to one. Hence, for example, with respect to the ICT-taxonomy we get:

Similarity (C++, Java) = 3/4
Similarity (C++, SQL) = 2/4 = 1/2
Similarity (Web, Semantic Web) = 1/2

Note that *Similarity (Ci,Cj)=0*, if *Ci* and *Cj* are not included in the same taxonomy. In such a case *Ci* and *Cj* do not have common ancestors, i.e., *CommonAncestors(Ci,Cj)=0.*

From technology point of view, the similarities of the concepts of each taxonomy can be easily precomputed and stored in matrixes. The size of each matrix is *n x n*, where *n* is the number of the nodes (concepts) in the taxonomy. Based on human knowledge the precomputed values of the matrixes can be manually changed if a need arises.

For now, we have only analyzed the similarities between two concepts. However, our ultimate goal is to compute the similarities of profiles such as the closeness of John's and Mary's FOAF profiles.

By closeness of two persons, say John and Mary, we denote by *Closeness(John, Mary),* and its value is the sum of the similarities of their *foaf:topic_interests*. That is:

$$Closeness\ (John, Mary) = \sum_{Ci\in A, Cj\in B} Similarity\ (Ci, Cj),$$

where *A* is the set of John's *foaf:topic_interest profile* and *B* is the set of Mary's *foaf:topic_interest profile*.

Note that *Closeness(John, Mary)=0,* if John and Mary do not have any common ancestors in their interests.

Closeness(John, Mary) has its maximal value when John's *foaf:topic_interest profile* is included in Mary's *foaf:topic_interest profile*. In such a case $Closeness\ (John, Mary)$ equals with the number of the items in John's *foaf:topic_interest* profile.

5. Weighted Profiles

5.1. Extending taxonomy-based semantic similarity by weights

So far we have assumed that a person is equally interested in all the topics he or she is interested in. However, in reality, the intensiveness of persons' interests varies: for example, an interest may be high, average or minor. Hence,

by including such features in users' profiles we could increase the expression power of person' profiles as well as queries on profiles.

We now extend our information retrieval model by weights. However, weighting of interests is beyond the expression power of *foaf:topic_interest* property. Therefore we have extended the expression power of the Taxonomy Ontology presented in Fig. 2. The Weighted Taxonomy ontology (Fig. 6.) allows attaching weights for interests: each instance of the class *weighted_interest* connects a weight to a person and a taxonomy item.

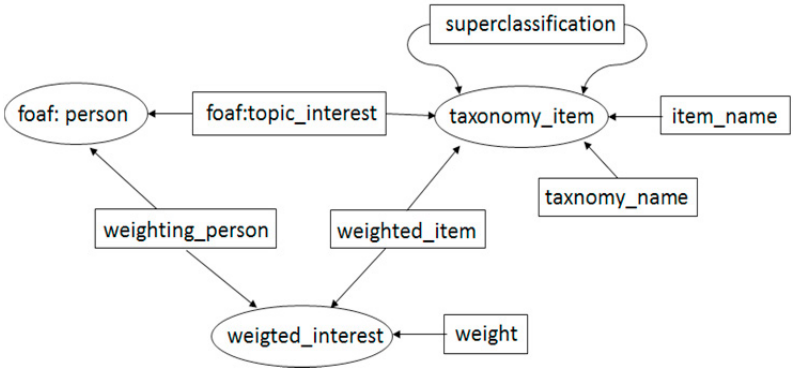

Fig. 6. Graphical presentation of the Weighted Taxonomy Ontology.

In specifying a weighted profile a person marks those nodes that match to his topics of interests and gives weights to those nodes. In order to simplify weighting, we have introduced three textual weights, namely *high*, *medium* and *low*. Their computational numerical values are 1.0, 0.50 and 0.25, respectively. For example, John might set weight *low* to node 'Java', weight *medium* to node 'RDF', and weight *high* to node 'Personal health record'.

In computing weighted similarity of the two distinct concepts *Ci* and *Cj*, we multiply *Similarity (Ci, Cj)* by the weights of these concepts. That is:

$$WeightedSimilarity\ (Wi(Ci), Wj(Cj)) = Wi \times Wj \times Similarity(Ci, Cj),$$

where *Wi* is the weight of concept *Ci* and *Wj* is the weight of concept *Cj*.

Furthermore, we define the *weighted closeness* of two persons, say John and Mary, as follows:

$$WeightedCloseness\ (John, Mary) = \sum_{C1 \in A, C2 \in B} WeightedSimilarity\ (W1(C1), W2(C2)),$$

where *A* is the set of John's *foaf:topic_interests in his profile,* and *B* is the set of Mary's *foaf:topic_interests in her profile*.

5.2. Relevancy ranking of persons' profiles

The Vector Model is an algebraic model for representing text documents as vectors of identifiers, such as, for example, taxonomy items[19]. The Vector Model is used in information filtering, information retrieval, indexing and relevancy rankings[20]. In this model, documents and queries are represented as vectors. Each dimension corresponds to a separate identifier or taxonomy item. So we can more accurately specify the queries and the contents of the documents, such as persons' profiles. Further, we can process the query by computing the distance of the query vector and the profile vectors.

This kind of computing requires that the sum of the weights of each vector equals to a predefined constant. For convenience, the used constant is usually one. Further, computing the distance requires that the vectors be specified in an orthogonal vector space[21]. This means that the leave nodes of the taxonomy should be independent. Therefore

we cannot use both parent and a child node as dimensions because they represent a subclassification relation, i.e., they are not independent. Instead leave nodes are usually independent

In the result of the query, the profiles are sorted in the order determined by the relevancy (similarity), i.e. the profile having the best match with the query is presented first. The number of the documents in the result can be restricted by requiring a certain degree of similarity.

In computing the closeness of John's and Mary's profile, we have to specify their profiles as a vector in a vector space determined by a taxonomy or taxonomies, which leaves are independent among themselves. For example we might use the three taxonomies of Fig. 1. In such a case the leaves nodes of the three taxonomies comprise the dimensions of the used vector space. Hence, we can query profiles that have as similar profile as possible with Joh's profile as follows:

$$< 0.25 \times \text{Java}, 0.5 \times \text{RDF}, 0.25 \times \text{Personal health record}, 0 \times \text{Punk}, \ldots. >$$

Each dimension is weighted in a way that the sum of the weights equals to one. In this case, the weights of the three first vectors have values, which are unequal to zero while all the other weights equal to zero.

5.3. Matching Algorithms

We have also investigated relevancy rankings by analyzing the different matching algorithms[14] (i.e., algorithms computing the distances between vectors). Initially we tested these algorithms in the context news items[22] and learning objects[23], where we compute the distance of user profiles and news items or learning objects.

We next give a short description of our tested algorithms. The Cosine matching algorithm[9] calculates the cosine measure between the query vector and the profiles. As a matter of fact the algorithm does not compute distance measures but rather approximates distance measures by computing the angles of the query vector and the vectors representing profiles. The Euclidean matching algorithm[20] calculates the Euclidean distance from the query vector to all profiles. The Manhattan distance algorithm[21]calculates so called "city block-distance". This measure indicates how many blocks in a city one would have to walk between two points.

The computing time for matching of each algorithm was performed for a file of 10 000 profiles. The differences of Euclidean, Cosine and Manhattan algorithms were rather small (less than 10 per cent), meaning that all these algorithms are suitable for computing the closeness of profiles.

6. Conclusions

The social network models and the Semantic Web support each other: social net-work provides a new paradigm for knowledge management in which users outsource knowledge via social network while Semantic Web enables the presentation of social information in a machine understandable form. FOAF, Dublin Core, and vCard are examples of popular vocabularies that provide vocabularies for knowledge representation. These vocabularies have proven to be useful and workable innovation. However, there are still many challenging issues concerning knowledge representation and knowledge management in social networks.

In this paper, we have increased the semantics of semantic social networks in two ways: by exploiting the semantics of taxonomies in measuring the similarity of user profiles, and by attaching weights into individuals' interests. The critical issue here, as well as in any approach in capturing additional semantics, is that they complicate the processing and management of social information.

Another problem is that some ontologies are dynamic: as the domain of the ontology evolves also the ontology must be updated accordingly, i.e., ones new concepts emerge they must be included into the appropriate ontologies. Thus, a feature of some ontologies is that they are incremental by their nature. This in turn brings a new problem to information retrieval: how can we match users- queries against documents, if they are not based on the same ontology. In our future research, we will restrict ourselves on this problem.

References

1. N. Ellison, C., Lampe, and C. Steinfield. Social Network Sites and Society: Current Trends and Future Possibilities. Available at: https://www.msu.edu/~nellison/EllisonLampeSteinfield2009.pdf
2. G. Ereteo, F. Gandon, and M. Buffa. Semantic Social Network Analysis. Available at: http://arxiv.org/ftp/arxiv/papers/0904/0904.3701.pdf
3. L. Dodds. An Introduction to FOAF. Available at: http://www.xml.com/pub/a/2004/02/04/foaf.html
4. Dublin Core metadata Iniative. Available at: http://dublincore.org/documents/dces/
5. Representing vCard Objects in RDF. Available at: http://www.w3.org/Submission/vcard-rdf/
6. RDF – Resource Description Language. Available at: http://www.w3.org/RDF/.
7. OWL – WEB Ontology Language. Available at: http://www.w3.org/TR/owl-features
8. FOAF Vocabulary Specification. Available at: http://xmlns.com/foaf/spec/
9. R. Baeza-Yates, and B. Ribeiro-Neto, B. Modern Information Retrieval (Addison Wesley, 1999).
10. SPARQL: Query Language for RDF. Available at: http://www.w3.org/TR/rdf-sparql-query/.
11. M. Daconta, L. Obrst, and K. Smith. The semantic web. Indianapolis: John Wiley & Sons. 2003.
12. T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing. Padua workshop on Formal Ontology, March 1993.
13. J. Davies, D. Fensel, and F. Harmelen, F. Towards the semantic web: ontology driven knowledge management. West Sussex: John Wiley & Sons.2002
14. G. Antoniou and F. Harmelen. A semantic web primer. The MIT Press. 2004.
15. Turtle - Terse RDF Triple Language. Available at: http://www.w3.org/TeamSubmission/turtle/.
16. DuCharme, 2011. Learning SPARQL. O'Reilly Media.
17. D. Gilles, et al. DYMAMicOntology for Information Retrieval. Available at: https://www.irit.fr/publis/SIG/2010_RIAO_DHMMRTRMMLR.pdf
18. D. Lin. Information-theoretic definition of similarity. 15 th International Conference on Machine Learning. 1998.
19. A. Kumar, A. Gupta, M. Batool, and S. Trehan. Latent Semantic Indexing-Based Intelligent Information Retrieval System for Digital Libraries. Journal of Computing and Information Technology - CIT 14, 2006, 3, 191–196
20. J. Friedman et al, An algorithm for finding best matches in logaritmic expected time, ACM Transactions on Mathematical Software, 3 (3), 1977, 209-226.
21. J. Bentley, B. Weide, and A. Ya, Optimal expected-time algorithms for closest point problem, ACM Transactions on Mathematical Software, 6(4), 1980, 563-580.
22. J.Yli-Koivisto, and J. Puustjärvi, CoMet: An Electronic Newspaper Prototype. In Proc. Of the Eight International Conference on Distributed Multimedia Systems, 2002, 703-707.
23. J. Puustjärvi. Integrating e-Learning Systems. In the Proc. of the IASTED International Conference on Web-Based Education (WBE 2004), Pages 417-421, 2004.